**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**VIỆN KỸ THUẬT – CÔNG NGHỆ**



**ĐỒ ÁN MÔN HỌC  
TRÍ TUỆ NHÂN TẠO**

**NHẬN DIỆN BIỂN SỐ XE BẰNG K-NEAREST NEIGHBORS**

**GVHD: ThS.Nguyễn Hải Vĩnh Cường**

|  |  |
| --- | --- |
| **SVTH:**  **Đỗ Đăng Khoa** | **1924801030198** |
| **Trương Quốc Hưng**  **Lớp: D19PM03** | **1924801030004** |

**BÌNH DƯƠNG – 10/2022**

**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**VIỆN KỸ THUẬT – CÔNG NGHỆ**



**ĐỒ ÁN MÔN HỌC  
TRÍ TUỆ NHÂN TẠO**

**NHẬN DIỆN BIỂN SỐ XE BẰNG K-NEAREST NEIGHBORS**

**GVHD: ThS.Nguyễn Hải Vĩnh Cường**

|  |  |
| --- | --- |
| **SVTH:**  **Đỗ Đăng Khoa** | **1924801030198** |
| **Trương Quốc Hưng**  **Lớp: D19PM03** | **1924801030004** |

**BÌNH DƯƠNG – 10/2022**

**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**KHOA KỸ THUẬT – CÔNG NGHỆ**

**NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN**

Họ và tên giảng viên: **Ths. Nguyễn Hải Vĩnh Cường**

Tên đề tài:

Nội dung nhận xét:

**Điểm:**

Bằng số:

Bằng chữ:

|  |  |
| --- | --- |
|  | **GIẢNG VIÊN CHẤM**  *(Ký, ghi rõ họ tên)* |

**ThS. Nguyễn Hải Vĩnh Cường**

**LỜI CẢM ƠN**

Lời đầu tiên, nhóm em xin bày tỏ sự cảm ơn chân thành đối với thầy Th.S Nguyễn Hải Vĩnh Cường – giáo viên hướng dẫn trực tiếp của nhóm em. Thầy Cường đã cho nhóm em những gợi ý và chỉ dẫn quý báu trong quá trình nghiên cứu và hoàn thiện đồ án này.

Mặc dù đã cố gắng hoàn thành đồ án nhưng chắc chắn sẽ không tránh khỏi những sai sót, nhóm em kính mong nhận được sự thông cảm và chỉ bảo của thầy và các bạn.

Nhóm em xin chân thành cảm ơn!

**TÓM TẮT**

Hiện nay, số lượng xe cộ tham gia giao thông trên đường là rất lớn dẫn đến tiêu tốn rất nhiều nhân lực và vật lực cho việc quản lý phương tiện cá nhân trong bãi gửi xe. Nếu không có một công cụ thuận tiện thì việc quản lý phương tiện cá nhân rất mất thời gian, dễ gây nhầm lẫn, thiệt hay cho người sử dụng dịch vụ tại các bãi đỗ xe.

Để giảm tải cho các công việc như thu tiền, bảo hiểm xe, tìm xe cộ trong bãi đỗ xe, trên thế giới đã phát triển công nghệ giám sát tự động đối với các phương tiện giao thông, chính nhờ tính cá nhân của biển số xe mà nó đã trở thành đối tượng chính được sử dụng để nghiên cứu, phát triển trong công nghệ này.

Do đó em muốn chọn đề tài này như bước căn bản trong việc tìm hiểu các công cụ giám sát mạnh hơn như kiểm soát xe lưu thông trên đường hay nhận dạng khuôn mặt ... đang được thế giới rất chú trọng lúc này.

Cấu trúc của bài báo cáo được chia thành 4 phần. Chương 1: Khái quát về đồ án, Chương 2: Cơ sở lý thuyết, Chương 3: Quy trình thực hiện và sản phẩm, Chương 4 : Kết luận.

**SUMMARY**

The cake classification application is made after collecting and surveying information and applying the Support Vector Machine algorithm. Based on python as the main language of the application.

The structure of the report is divided into four parts. Chapter 1: Overview of the project, Chapter 2: Theoretical background, Chapter 3: Implementation process and products, Chapter 4: Conclusion.

**MỤC LỤC**

[CHƯƠNG I: KHÁI QUÁT VỀ ĐỒ ÁN 1](#_Toc117240196)

[1.1.Tên đề tài 1](#_Toc117240197)

[“Nhận diện biển số xe bằng K-Nearest Neighbors”. 1](#_Toc117240198)

[1.2.Tổng quan 1](#_Toc117240199)

[1.3.Nhiệm vụ đề tài 1](#_Toc117240200)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 2](#_Toc117240201)

[2.1.Giới thiệu về thuật toán K-Nearest Neigbors 2](#_Toc117240202)

[2.1.1.K-Nearest Neigbors là gì? 2](#_Toc117240203)

[2.2.Ưu Điểm của K-Nearest Neigbors 2](#_Toc117240204)

[2.3.Nhược điểm của K-Nearest Neigbors 2](#_Toc117240205)

[2.4.Các ngôn ngữ sử dụng trong đồ án 2](#_Toc117240206)

[2.4.1.Ngôn ngữ lập trình Python 2](#_Toc117240207)

[CHƯƠNG 3: QUY TRÌNH THỰC HIỆN VÀ SẢN PHẨM 6](#_Toc117240208)

[3.1.Khái niệm biển số xe 6](#_Toc117240209)

[3.2.Xử lý ảnh và Open CV 6](#_Toc117240210)

[3.3.Hướng giải quyết 7](#_Toc117240211)

[3.4.Phát hiện vị trí và tách biển số xe 8](#_Toc117240212)

[3.4.1.Hướng giải quyết 8](#_Toc117240213)

[3.5.Chuyển ảnh xám 9](#_Toc117240214)

[3.6.Tăng độ tương phản 9](#_Toc117240215)

[3.6.1.Phép toán hình thái học 9](#_Toc117240216)

[3.6.2.Tăng độ tương phản 12](#_Toc117240217)

[3.7.Giảm nhiễu bằng bộ lọc Gauss 12](#_Toc117240218)

[3.7.1.Nhiễu 12](#_Toc117240219)

[3.7.2.Bộ lọc Gauss (Gauss filter) 13](#_Toc117240220)

[3.8.Nhị phân hóa với ngưỡng động (Adaptive Threshold) 14](#_Toc117240221)

[3.8.1.Ảnh nhị phân 14](#_Toc117240222)

[3.8.2.Nhị phân hóa 14](#_Toc117240223)

[3.8.3.Nhị phân hóa với ngưỡng động 15](#_Toc117240224)

[3.9.Phát hiện cạnh Canny (Canny Edge Detection) 16](#_Toc117240225)

[3.10.Lọc biển số với contour 19](#_Toc117240226)

[3.10.1.Một số phương pháp tìm contour 19](#_Toc117240227)

[3.10.2.Lọc biển số 22](#_Toc117240228)

[3.11.Phân đoạn kí tự 23](#_Toc117240229)

[3.11.1. Hướng giải quyết 23](#_Toc117240230)

[3.11.2.Xoay biển số 24](#_Toc117240231)

[3.11.3.Tìm vùng đối tượng 25](#_Toc117240232)

[3.11.4.Tìm và tách kí tự 26](#_Toc117240233)

[CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 27](#_Toc117240234)

[4.1.Kết luận 27](#_Toc117240235)

[4.2.Hướng phát triển 27](#_Toc117240236)

# CHƯƠNG I: KHÁI QUÁT VỀ ĐỒ ÁN

## 1.1.Tên đề tài

## “Nhận diện biển số xe bằng K-Nearest Neighbors”.

## 1.2.Tổng quan

Nội dung :

* + Tìm hiểu về biển số xe và hệ thống nhận dạng biển số xe
  + Phát biểu bài toán và hướng giải quyết
  + Nghiên cứu một số thuật toán xử lý ảnh và nhận dạng kí tự ứng dụng trong việc nhận dạng biển số xe

## 1.3.Nhiệm vụ đề tài

Từ nội dung nêu trên, đề tài của em sẽ bao gồm các nhiệm vụ sau:

* + Tìm hiểu khái quát về xử lý ảnh và bài toán nhận dạng biển số xe.
  + Tìm hiểu thông tin về biển số xe và phân loại biển số xe của Việt Nam.
  + Tìm hiểu các công đoạn chính của bài toán nhận dạng biển số xe gồm 3 khâu chính:
    - Phát hiện vị trí và tách biển số xe
    - Phân đoạn kí tự trong biển số xe
    - Nhận dạng kí tự
  + Cài đặt thử nghiệm.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1.Giới thiệu về thuật toán K-Nearest Neigbors

### 2.1.1.K-Nearest Neigbors là gì?

KNN (K-Nearest Neighbors) là một trong những thuật toán học có giám sát đơn giản nhất được sử dụng nhiều trong khai phá dữ liệu và học máy. Ý tưởng của thuật toán này là nó không học một điều gì từ tập dữ liệu học (nên KNN được xếp vào loại lazy learning), mọi tính toán được thực hiện khi nó cần dự đoán nhãn của dữ liệu mới.  
      Lớp (nhãn) của một đối tượng dữ liệu mới có thể dự đoán từ các lớp (nhãn) của k hàng xóm gần nó nhất.

## 2.2.Ưu Điểm của K-Nearest Neigbors

* Thuật toán đơn giản, dễ dàng triển khai.
* Độ phức tạp tính toán nhỏ.
* Xử lý tốt với tập dữ liệu nhiễu

## 2.3.Nhược điểm của K-Nearest Neigbors

1. Với K nhỏ dễ gặp nhiễu dẫn tới kết quả đưa ra không chính xác
2. Cần nhiều thời gian để thực hiện do phải tính toán khoảng cách với tất cả các đối tượng trong tập dữ liệu.
3. Cần chuyển đổi kiểu dữ liệu thành các yếu tố định tính.

## 2.4.Các ngôn ngữ sử dụng trong đồ án

### 2.4.1.Ngôn ngữ lập trình Python

#### 2.4.1.1.Python là gì?

[Python](https://quantrimang.com/python) là ngôn ngữ lập trình hướng đối tượng, cấp cao, mạnh mẽ, được tạo ra bởi Guido van Rossum. Nó dễ dàng để tìm hiểu và đang nổi lên như một trong những ngôn ngữ lập trình nhập môn tốt nhất cho người lần đầu tiếp xúc với ngôn ngữ lập trình. Python hoàn toàn tạo kiểu động và sử dụng cơ chế cấp phát bộ nhớ tự động. Python có [cấu trúc dữ liệu](https://quantrimang.com/cau-truc-du-lieu-va-giai-thuat) cấp cao mạnh mẽ và cách tiếp cận đơn giản nhưng hiệu quả đối với lập trình hướng đối tượng. Cú pháp lệnh của Python là điểm cộng vô cùng lớn vì sự rõ ràng, dễ hiểu và cách gõ linh động làm cho nó nhanh chóng trở thành một ngôn ngữ lý tưởng để viết script và phát triển ứng dụng trong nhiều lĩnh vực, ở hầu hết các nền tảng.

Python là một ngôn ngữ khá cũ được tạo ra bởi Guido Van Rossum. Thiết kế bắt đầu vào cuối những năm 1980 và được phát hành lần đầu tiên vào tháng 2 năm 1991.

Vào cuối những năm 1980, Guido Van Rossum làm việc trong Amoeba, phân phối một nhóm hệ điều hành. Ông muốn sử dụng một ngôn ngữ thông dịch như ABC (ABC có cú pháp rất dễ hiểu) để truy cập vào những cuộc gọi hệ thống Amoeba. Vì vậy, ông quyết định tạo ra một ngôn ngữ mở rộng. Điều này đã dẫn đến một thiết kế của ngôn ngữ mới, chính là Python sau này.

* + 1. **Tính năng của Python**

**Ngôn ngữ lập trình đơn giản, dễ học:** Python có cú pháp rất đơn giản, rõ ràng. Nó dễ đọc và viết hơn rất nhiều khi so sánh với những ngôn ngữ lập trình khác như C++, Java, C#. Python làm cho việc lập trình trở nên thú vị, cho phép bạn tập trung vào những giải pháp chứ không phải cú pháp.

**Miễn phí, mã nguồn mở:** Bạn có thể tự do sử dụng và phân phối Python, thậm chí là dùng nó cho mục đích thương mại. Vì là mã nguồn mở, bạn không những có thể sử dụng các phần mềm, chương trình được viết trong Python mà còn có thể thay đổi mã nguồn của nó. Python có một cộng đồng rộng lớn, không ngừng cải thiện nó mỗi lần cập nhật.

**Khả năng di chuyển:** Các chương trình Python có thể di chuyển từ nền tảng này sang nền tảng khác và chạy nó mà không có bất kỳ thay đổi nào. Nó chạy liền mạch trên hầu hết tất cả các nền tảng như Windows, macOS, Linux.

**Khả năng mở rộng và có thể nhúng:** Giả sử một ứng dụng đòi hỏi sự phức tạp rất lớn, bạn có thể dễ dàng kết hợp các phần code bằng C, [C++](https://quantrimang.com/cplusplus) và những ngôn ngữ khác (có thể gọi được từ C) vào code Python. Điều này sẽ cung cấp cho ứng dụng của bạn những tính năng tốt hơn cũng như khả năng scripting mà những ngôn ngữ lập trình khác khó có thể làm được.

**Ngôn ngữ thông dịch cấp cao:** Không giống như [C](https://quantrimang.com/lap-trinh-c)/C++, với Python, bạn không phải lo lắng những nhiệm vụ khó khăn như quản lý bộ nhớ, dọn dẹp những dữ liệu vô nghĩa,... Khi chạy code Python, nó sẽ tự động chuyển đổi code sang ngôn ngữ máy tính có thể hiểu. Bạn không cần lo lắng về bất kỳ hoạt động ở cấp thấp nào.

**Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến:** Python có một số lượng lớn thư viện tiêu chuẩn giúp cho công việc lập trình của bạn trở nên dễ thở hơn rất nhiều, đơn giản vì không phải tự viết tất cả code. Ví dụ: Bạn cần kết nối [cơ sở dữ liệu](https://quantrimang.com/co-so-du-lieu) MySQL trên Web server? Bạn có thể nhập thư viện MySQLdb và sử dụng nó. Những thư viện này được kiểm tra kỹ lưỡng và được sử dụng bởi hàng trăm người. Vì vậy, bạn có thể chắc chắn rằng nó sẽ không làm hỏng code hay ứng dụng của mình.

**Hướng đối tượng:** Mọi thứ trong Python đều là hướng đối tượng. [Lập trình hướng đối tượng](https://quantrimang.com/steve-jobs-dinh-nghia-lap-trinh-huong-doi-tuong-khien-ca-the-gioi-than-phuc-131900) (OOP) giúp giải quyết những vấn đề phức tạp một cách trực quan. Với OOP, bạn có thể phân chia những vấn đề phức tạp thành những tập nhỏ hơn bằng cách tạo ra các đối tượng.

* + 1. **Python được dùng ở đâu?**

**Lập trình ứng dụng web:** Bạn có thể tạo web app có khả năng mở rộng (scalable) được bằng cách sử dụng framework và CMS (Hệ thống quản trị nội dung) được tích hợp trong Python. Vài nền tảng phổ biến để tạo web app là: Django, Flask, Pyramid, Plone, Django CMS. Các trang như Mozilla, Reddit, Instagram và PBS đều được viết bằng Python.

**Khoa học và tính toán:** Có nhiều thư viện trong Python cho khoa học và tính toán số liệu, như SciPy và NumPy, được sử dụng cho những mục đích chung chung trong tính toán. Và, có những thư viện cụ thể như: EarthPy cho khoa học trái đất, AstroPy cho Thiên văn học,... Ngoài ra, Python còn được sử dụng nhiều trong machine learning, khai thác dữ liệu và deep learning.

**Tạo nguyên mẫu phần mềm:** Python chậm hơn khi so sánh với các ngôn ngữ được biên dịch như C++ và Java. Nó có thể không phải là lựa chọn tốt nếu nguồn lực bị giới hạn và yêu cầu về hiệu quả là bắt buộc. Tuy nhiên, Python là ngôn ngữ tuyệt vời để tạo những nguyên mẫu (bản chạy thử - prototype). Ví dụ, bạn có thể sử dụng Pygame (thư viện viết game) để tạo nguyên mẫu game trước. Nếu thích nguyên mẫu đó có thể dùng C++ để viết game thực sự.

**Ngôn ngữ tốt để dạy lập trình:** Python được nhiều công ty, trường học sử dụng để dạy lập trình cho trẻ em và những người mới lần đầu học lập trình. Bên cạnh những tính năng và khả năng tuyệt vời thì cú pháp đơn giản và dễ sử dụng của nó là lý do chính cho việc này

# CHƯƠNG 3: QUY TRÌNH THỰC HIỆN VÀ SẢN PHẨM

## 3.1.Khái niệm biển số xe

Ở Việt Nam, biển kiểm soát xe cơ giới (hay còn gọi tắt là biển kiểm soát, biển số xe) là tấm biển gắn trên mỗi xe cơ giới, được cơ quan công an cấp (đối với xe quân sự do Bộ Quốc phòng cấp) khi mua xe mới hoặc chuyển nhượng xe. Biển số xe được làm bằng hợp kim nhôm sắt, có dạng hình chữ nhật hoặc hơi vuông, trên đó có in số và chữ (biển xe dân sự không dùng các chữ cái I, J, O, Q, W. Chữ R chỉ dùng cho xe rơ-moóc, sơ-mi rơ-moóc) cho biết: Vùng và địa phương quản lý, các con số cụ thể khi tra trên máy tính còn cho biết danh tính người chủ hay đơn vị đã mua nó, thời gian mua nó phục vụ cho công tác an ninh, đặc biệt trên đó còn có hình Quốc huy Việt Nam dập nổi.

Tiêu chuẩn về kích thước: Ở mỗi nước thường có tiêu chuẩn về kích thước nhất định, còn riêng Việt Nam tỉ lệ kích thước giữa các biển số là gần như giống nhau. Biển số xe có 2 loại, kích thước như sau: Loại biển số dài có chiều cao 110 mm, chiều dài 470 mm; loại biển số ngắn có chiều cao 200 mm, chiều dài 280 mm nên ta sẽ giới hạn tỉ lệ cao/rộng là

3.5 ≤ cao/rộng ≤ 6.5 (biển một hàng) và 0.8 ≤ cao/rộng ≤ 1.5 (biển hai hàng).

Số lượng kí tự trong biển số xe nằm trong khoảng [7,9]. Chiều cao của chữ và số: 80mm, chiều rộng của chữ và số: 40mm.

Từ những đặc điểm trên ta có thể thiết lập nhưng thông số, điều khiển để lọc chọn những đối tương phù hợp mà ta cần.

## 3.2.Xử lý ảnh và Open CV

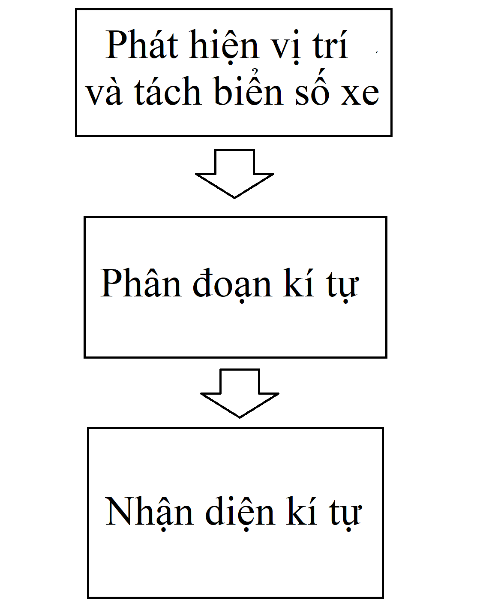
Xử lý ảnh là một phân ngành trong xử lý số tín hiệu với tín hiệu xử lý là ảnh. Đây là một phân ngành khoa học mới rất phát triển trong những năm gần đây. Xử lý ảnh gồm 4 lĩnh vực chính: xử lý nâng cao chất lượng ảnh, nhận dạng ảnh, nén ảnh và truy vấn ảnh. Sự phát triển của xử lý ảnh đem lại rất nhiều lợi ích cho cuộc sống của con người. Ngày nay xử lý ảnh đã được áp dụng rất rộng rãi trong đời sống như: photoshop, nén ảnh, nén video, nhận dạng biển số xe, nhận dạng khuôn mặt, nhận dạng chữ viết, xử lý ảnh thiên văn, ảnh y tế,....

OpenCV (Open Computer Vision) là một thư viện mã nguồn mở hàng đầu cho xử lý về thị giác máy tính, machine learning, xử lý ảnh. OpenCV đươc viết bằng C/C++, vì vậy có tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực. Opencv có các interface cho C/C++, Python Java vì vậy hỗ trợ được cho Window, Linux, MacOs lẫn Android, iOS OpenCV có cộng đồng hơn 47 nghìn người dùng và số lượng download vượt quá 6 triệu lần. Opencv có rất nhiều ứng dụng như:

* Nhận dạng ảnh
* Xử lý hình ảnh
* Phục hồi hình ảnh/video
* Thực tế ảo
* Các ứng dụng khác

## 3.3.Hướng giải quyết

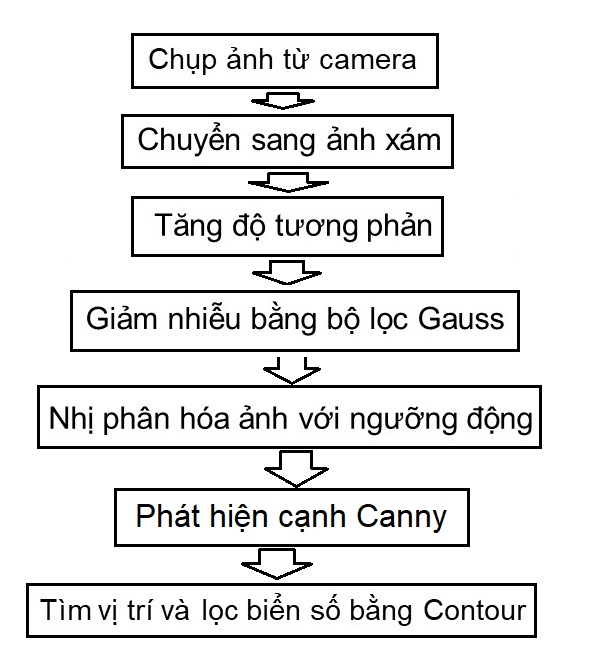
Hiện nay trên thế giới đã có rất nhiều cách tiếp cận khác nhau với việc nhận dạng biển số xe, tuy nhiên trong phạm vi đồ án này em sẽ giải quyết vấn đề theo 3 bước chính:

1. Phát hiện vị trí và tách biển số xe từ một hình ảnh có sẵn từ đầu vào là camera
2. Phân đoạn các kí tự có trong biển số xe
3. Nhận diện các kí tự đó rồi đưa về mã ASCII

## 3.4.Phát hiện vị trí và tách biển số xe

### 3.4.1.Hướng giải quyết

Sơ đồ dưới đây sẽ tóm gọn các bước để xác định và tách biển số xe:



Đầu tiên ta sẽ cắt từng frame ảnh ra từ đầu vào để xử lý, tách biển số. Ở phạm vi đồ án này, ý tưởng chủ yếu là nhận diện được biển số từ sự thay đổi đột ngột về cường độ ánh sáng giữa biển số và môi trường xung quanh nên ta sẽ loại bỏ các dữ liệu màu sắc RGB bằng cách chuyển sang ảnh xám. Tiếp theo ta tăng độ tương phản với hai phép toán hình thái học Top Hat và Black Hat để làm nổi bật thêm biển số giữa phông nền, hỗ trợ cho việc xử lý nhị phân sau này. Sau đó, ta giảm nhiễu bằng bộ lọc Gauss để loại bỏ những chi tiết nhiễu có thể gây ảnh hưởng đến quá trình nhận diện, đồng thời làm tăng tốc độ xử lý.

Việc lấy ngưỡng sẽ giúp ta tách được thông tin biển số và thông tin nền, ở đây em chọn lấy ngưỡng động (Adaptive Threshold). Tiếp đó ta sử dụng thuật toán phát hiện cạnh Canny để trích xuất những chi tiết cạnh của biển số. Trong quá trình xử lý máy tính có thể nhầm lẫn biển số với những chi tiết nhiễu, việc lọc lần cuối bằng các tỉ lệ cao/rộng hay diện tích của biển số sẽ giúp xác định được đúng biển số. Cuối cùng, ta sẽ xác định vị trí của biển số trong ảnh bằng cách vẽ Contour bao quanh.

## 3.5.Chuyển ảnh xám

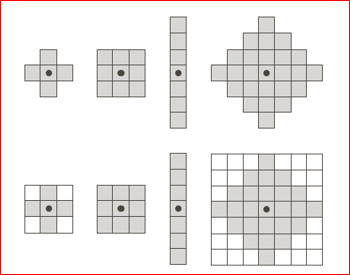
Ảnh xám (Gray Scale) đơn giản là một hình ảnh trong đó các màu là các sắc thái của màu xám với 256 cấp độ xám biến thiên từ màu đen đến màu trắng, nằm trong giải giá trị từ 0 đến 255, nghĩa là cần 8 bits hay 1 byte để biểu diễn mỗi điểm ảnh này. Lý do cần phải phân biệt giữa ảnh xám và các ảnh khác nằm ở việc ảnh xám cung cấp ít thông tin hơn cho mỗi pixel. Với ảnh thông thường thì mỗi pixel thường được cung cấp 3 trường thông tin trong khi với ảnh xám chỉ có 1 trường thông tin, việc giảm khối lượng thông tin giúp tăng tốc độ xử lý, đơn giản hóa giải thuật nhưng vẫn đảm bảo các tác vụ cần thiết

Ở bài này em sẽ chuyển ảnh xám từ hệ màu HSV thay vì RGB vì với không gian màu HSV ta có ba giá trị chính là: Vùng màu (Hue), độ bão hòa (Saturation), cường độ sáng (Value). Vì lý do đó không gian màu HSV thích nghi tốt hơn đối với sự thay đổi ánh sáng từ môi trường ngoài. Khi chuyển đổi, ảnh xám ta cần là ma trận các giá trị cường độ sáng tách ra từ hệ màu HSV.

## **3.6.Tăng độ tương phản**

### 3.6.1.Phép toán hình thái học

Hình thái học toán học là một lý thuyết và kỹ thuật để phân tích và xử lý cấu trúc hình học, hình ảnh đầu ra được xác định chủ yếu dựa vào phần từ cấu trúc (structuring elements/kernel)



Hình thái học toán học đã được phát triển cho hình ảnh nhị phân, và sau đó được mở rộng cho ảnh xám,... Đây là một trong những kỹ thuật được áp dụng trong giai đoạn tiền xử lý. Hai phép toán thường dùng là phép giãn nở (Dilation) và phép co (Erosion) . Từ hai phép toán cơ bản này người ta phát triển thành một số phép toán như phép đóng (Closing) và phép mở (Opening) và phép Top Hat, Black Hat.

1. Phép co

Phép toán co có ứng dụng trong việc giảm kích thước của đối tượng, tách rời các đối tượng gần nhau, làm mảnh và tìm xương của đối tượng.

|  |  |
| --- | --- |
|  |  |
| Ảnh gốc | Ảnh sau khi dùng phép co |

1. Phép giãn nở

Phép toán này có tác dụng làm cho đối tượng ban đầu trong ảnh tăng lên về kích thước (giãn nở ra). Ứng dụng của phép giãn nở là làm cho đối tượng trong ảnh được tăng lên về kích thước, các lỗ nhỏ trong ảnh được lấp đầy, nối liền đường biên ảnh đối với những đoạn rời nhỏ.

|  |  |
| --- | --- |
|  |  |
| Ảnh gốc | Ảnh sau khi dùng phép giãn nở |

1. Phép mở

Là thực hiện phép co trước sau đó mới thực hiện phép giãn nở. Phép toán mở được ứng dụng trong việc loại bỏ các phần lồi lõm và làm cho đường bao các đối tượng trong ảnh trở nên mượt mà hơn.

|  |  |
| --- | --- |
|  |  |
| Ảnh gốc | Ảnh sau khi dùng phép mở |

1. Phép đóng

Thực hiện phép giãn nở trước sau đó mới thực hiện phép co. Phép toán đóng được dùng trong ứng dụng làm trơn đường bao các đối tượng, lấp đầy các khoảng trống biên và loại bỏ những hố nhỏ.

|  |  |
| --- | --- |
|  |  |
| Ảnh gốc | Ảnh sau khi dùng phép đóng |

1. Phép Top Hat

Phép Top Hat là kết quả của phép trừ ảnh của ảnh ban đầu với ảnh sau khi thực hiện phép mở, dùng để làm nổi bật nhưng chi tiết trắng trong nền tối

|  |  |
| --- | --- |
|  |  |
| Ảnh gốc | Ảnh sau khi dùng phép Top Hat |

1. Phép Black Hat

Phép Black Hat là kết quả của phép trừ ảnh của ảnh sau khi thực hiện phép đóng với ảnh ban đầu. Dùng làm nổi bật chi tiết tối trong nền trắng.

|  |  |
| --- | --- |
|  |  |
| Ảnh gốc | Ảnh sau khi dùng phép Black Hat |

### 3.6.2.Tăng độ tương phản

Để làm tăng độ tương phản của biển số, em sử dụng chủ yếu hai phép Top Hat và Black Hat. Ý tưởng chung là ảnh đầu ra sẽ là ảnh gốc cộng thêm ảnh qua phép Top Hat và trừ đi ảnh qua phép Black Hat. Những chi tiết đã sáng sẽ sáng hơn và những chi tiết tối lại càng tối hơn, từ đó sẽ làm tăng độ tương phản cho biển số.

|  |  |
| --- | --- |
|  |  |
| Ảnh gốc | Ảnh sau khi tăng độ tương phản |

## 3.7.Giảm nhiễu bằng bộ lọc Gauss

### 3.7.1.Nhiễu

Noise được hiểu cơ bản là các dạng chấm hạt nhỏ phân bố trên hình ảnh. Noise có thể làm biến dạng các chi tiết trong ảnh khiến cho chất lượng ảnh thấp.

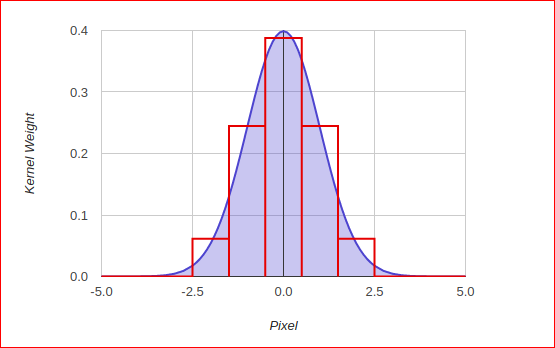
Trên thực tế có nhiều loại nhiễu, nhưng người ta thường chia làm ba loại: nhiễu cộng, nhiễu nhân và nhiễu xung. Bản chất của nhiễu thường tương ứng với tần số cao và cơ sở lý thuyết của bộ lọc là chỉ cho những tín hiệu có tần số nhất định đi qua, nên người ta thường sử dụng bộ lọc thông thấp hay trung bình.



### 3.7.2.Bộ lọc Gauss (Gauss filter)

Bộ lọc Gauss được cho là bộ lọc hữu ích nhất, được thực hiện bằng cách nhân chập ảnh đầu vào với một ma trận lọc Gauss sau đó cộng chúng lại để tạo thành ảnh đầu ra.

Ý tưởng chung là giá trị mỗi điểm ảnh sẽ phụ thuộc nhiều vào các điểm ảnh ở gần hơn là các điểm ảnh ở xa. Trọng số của sự phụ thuộc được lấy theo hàm Gauss (cũng được sử dụng trong quy luật phân phối chuẩn).



Giả sử ảnh là một chiều. Điểm ảnh ở trung tâm sẽ có trọng số lớn nhất. Các điểm ảnh ở càng xa trung tâm sẽ có trọng số giảm dần khi khoảng cách từ chúng tới điểm trung tâm tăng lên. Như vậy điểm càng gần trung tâm sẽ càng đóng góp nhiều hơn vào giá trị điểm trung tâm.

|  |  |
| --- | --- |
|  |  |
| Ảnh gốc | Ảnh sau khi làm mờ, giảm nhiễu |

## 3.8.Nhị phân hóa với ngưỡng động (Adaptive Threshold)

### 3.8.1.Ảnh nhị phân

Là ảnh mà giá trị của các điểm ảnh chỉ được biểu diễn bằng hai giá trị là 0 (Đen) và 255 (Trắng)

### 3.8.2.Nhị phân hóa

Là quá trình biến đổi một ảnh xám thành ảnh nhị phân.

* + Gọi giá trị cường độ sáng tại một điểm ảnh là I(x,y) .
  + INP(x,y) là cường độ sáng của điểm ảnh trên ảnh nhị phân .
  + (Với 0 < x < image.width) và (0 < y < image.height).

Để biến đổi ảnh xám thành ảnh nhị  phân. Ta so sánh giá trị cường độ sáng của điểm ảnh với một ngưỡng nhị phân **T**.

* + Nếu I(x,y) > **T** thì INP(x, y) = 0.
  + Nếu I(x,y) > **T** thì INP(x, y) = 255.

### 3.8.3.Nhị phân hóa với ngưỡng động

Việc nhị phân hóa ảnh với ngưỡng toàn cục như thông thường sẽ rất khó khăn khi phải tự tính toán và chọn mức ngưỡng phù hợp cho từng ảnh khác nhau. Nhị phân hóa ảnh ngưỡng động sẽ giúp tính toán ngưỡng cho phù hợp với từng ảnh, lợi thế thứ hai chính là nó rất phù hợp khi ảnh có vùng bị quá chói hoặc quá tối dẫn đến có thể mất luôn hình ảnh tại vùng đó nếu sử dụng ngưỡng toàn cục.

Về ý tưởng chính sẽ theo 3 bước sau:

1. Chia tấm ảnh thành nhiều khu vực, cửa sổ (Region) khác nhau.
2. Dùng một thuật toán để tìm một giá trị T phù hợp với từng cửa sổ.
3. Áp dụng phương pháp nhị phân hóa cho từng khu vực, cửa sổ với ngưỡng T phù hợp.

Có rất nhiều phương pháp để tìm T, ở đây em sử dụng một kiểu thuật toán mà thư viện OpenCV hỗ trợ là ADAPTIVE\_THRESH\_GAUSSIAN\_C tức lấy trung bình các giá trị xung quanh điểm ngưỡng động đang xét T(x,y) với phân phối Gauss rồi trừ đi hằng số C.



## 3.9.Phát hiện cạnh Canny (Canny Edge Detection)

Trong hình ảnh, thường tồn tại các thành phần như: vùng trơn, góc/cạnh và nhiễu. Cạnh trong ảnh mang đặc trưng quan trọng, thường là thuộc đối tượng trong ảnh. Do đó, để phát hiện cạnh trong ảnh, có nhiều giải thuật khác nhau như toán tử Sobel, toán tử Prewitt, Zero crossing .... nhưng ở đây em chọn giải thuật Canny vì hương pháp này hơn hẳn các phương pháp khác do ít bị tác động của nhiễu và cho khả năng phát hiện các biên yếu. Phương pháp này đi theo 4 bước chính:

1. Giảm nhiễu (Noise reduction)
2. Tính toán Gradient (Gradient calculation)
3. Loại bỏ những điểm không phải là cực đại (Non-maximum suppression)
4. Lọc ngưỡng (Double threshold)
5. Giảm nhiễu

Làm mờ ảnh, giảm nhiễu dùng bộ lọc Gauss kích thước 5x5. Kích thước 5x5 thường hoạt động tốt cho giải thuật Canny

1. Tính toán Gradient

Ta dùng 2 bộ lọc Sobel X và Sobel Y (3x3) để tính đạo hàm Gx và Gy

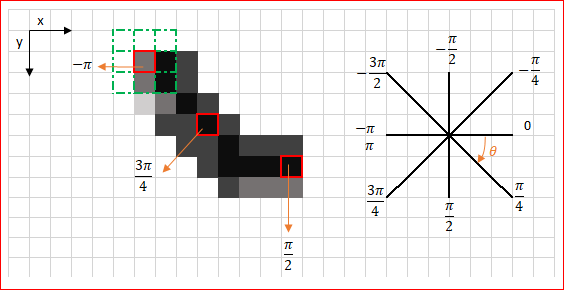
|  |  |
| --- | --- |
|  |  |

Tìm gradient và hướng được làm tròn về 4 hướng: hướng ngang (0 độ), hướng chéo bên phải (45 độ), hướng dọc (90 độ) và hướng chéo trái (135 độ).

|  |  |
| --- | --- |
|  |  |

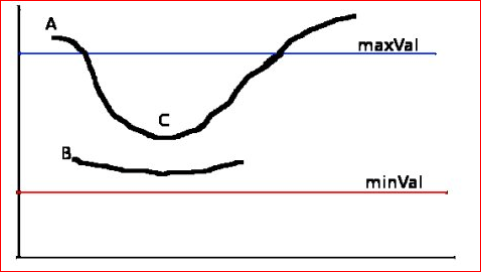
1. Loại bỏ những điểm không phải là cực đại

Ở bước này, dùng một filter 3x3 lần lượt chạy qua các pixel trên ảnh gradient. Trong quá trình lọc, xem xét xem độ lớn gradient của pixel trung tâm có phải là cực đại so với các gradient ở các pixel xung quanh. Nếu là cực đại, ta sẽ ghi nhận sẽ giữ pixel đó lại. Còn nếu pixel tại đó không phải là cực đại lân cận, ta sẽ set độ lớn gradient của nó về zero. Ta chỉ so sánh pixel trung tâm với 2 pixel lân cận theo **hướng gradient.** Ví dụ: nếu hướng gradient đang là 0 độ, ta sẽ so pixel trung tâm với pixel liền trái và liền phải nó. Trường hợp khác nếu hướng gradient là 45 độ, ta sẽ so sánh với 2 pixel hàng xóm là góc trên bên phải và góc dưới bên trái của pixel trung tâm



1. Lọc ngưỡng

**Lọc ngưỡng: ta sẽ xét các pixel dương trên mặt nạ nhị phân kết quả của bước trước. Nếu giá trị gradient vượt ngưỡng max\_val thì pixel đó chắc chắn là cạnh. Các pixel có độ lớn gradient nhỏ hơn ngưỡng min\_val sẽ bị loại bỏ. Còn các pixel nằm trong khoảng 2 ngưỡng trên sẽ được xem xét rằng nó có nằm liền kề với những pixel được cho là "chắc chắn là cạnh" hay không. Nếu liền kề thì ta giữ, còn không liền kề bất cứ pixel cạnh nào thì ta loại. Sau bước này ta có thể áp dụng thêm bước hậu xử lý loại bỏ nhiễu (tức những pixel cạnh rời rạc hay cạnh ngắn) nếu muốn**



1. Kết quả

Sau khi sử dụng phát hiện biên canny, dù đã trích xuất được những chi tiết cạnh của biển số, tuy nhiên vẫn còn quá nhiều chi tiết thừa trong hình ảnh, từ đây chúng ta sẽ vẽ contour, áp dụng nhưng đặc điểm của biển số để lọc lấy ra biển số chính xác.



## 3.10.Lọc biển số với contour

### 3.10.1.Một số phương pháp tìm contour

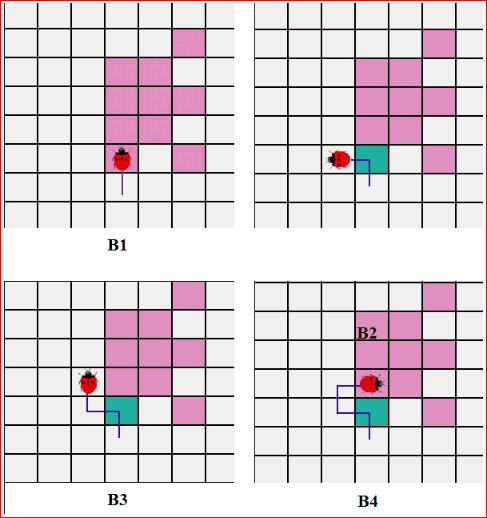
Có thể hiểu Contour là tập hợp các điểm tạo thành đường cong kín bao quanh một đối tượng nào đó. Thường dùng để xác định vị trí, đặc điểm của đối tượng. Có 4 thuật toán Contour Tracing chung nhất. Hai trong số đó có tên là: Square Tracing algorithm và Moore – Neighbor Tracing là dễ để thực hiện và thường xuyên được dùng để dò tìm contour của một mẫu. Với thư viện OpenCV người ta áp dụng thuật toán Suzuki’s Contour tracing. Dưới đây em sẽ trình bày kĩ hơn về 3 phương pháp trên:

1. Thuật toán Square Tracing

Duyệt từ pixel ngoài cùng bên trái phía dưới, đi lên cho tới khi gặp pixel có giá trị bằng 255 (pixel này sẽ được gọi là pixel start) thì bắt đầu di chuyển theo quy tắc sau:

* Nếu gặp Pixel có giá trị bằng 255 thì rẽ trái.
* Nếu gặp Pixel có giá trị bằng 0 thì rẽ phải.
* Di chuyển cho tới khi quay lại pixel start thì dừng lại.

Hình sau mô tả cách hoạt động của thuật toán:

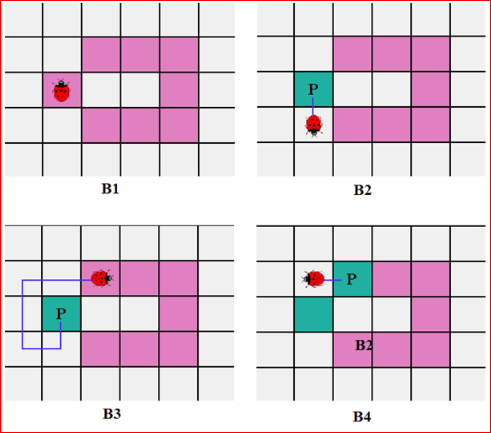


Thuật toán sẽ kết thúc đúng khi di chuyển vào pixel start lần thứ 2 sau khi đi qua n pixel khác và theo đúng hướng đi vào pixel start lần đầu tiên. Và sai khi di chuyển vào pixel start mà không đúng hướng ban đầu. Vậy thuật toán này chỉ chạy đúng trên đối tượng   
4 - connected.

|  |  |
| --- | --- |
|  |  |

1. Thuật toán Moore – Neighbor

Thuật toán này có chút khác biệt so với thuật toán Square Tracking, cụ thể: khi gặp pixel có giá trị bằng 255 đầu tiên (pixel start) thì ta sẽ quay lại pixel trước đó, sau đó đi vòng qua các pixel thuộc 8-connected theo chiều kim đồng hồ cho tới khi gặp pixel khác có giá trị bằng 255. Và điệu kiện kết thúc cũng giống như thuật toán Square Tracking



1. Thuật toán Suzuki’s Tracing

Đây là thuật toán được thư viện OpenCV sử dụng, ngoài khả năng xác định được biên của vật thể như hai phương pháp trên. Phương pháp Suzuki’s Tracing còn có khả năng phân biệt được đường biên ngoài (Outer) và đường biên trong (Hole) của vật thể.

Hàm trong OpenCV được biểu diễn như sau:

**findContours**(InputOutputArray image, OutputArrayOfArrays contours, OutputArray hierarchy, int mode, int method, Point offset=Point())

Các tham số:   
**image** : hình ảnh cần tìm biên, là ảnh nhị phân.  
**contours** : lưu trữ các đường biên tìm được, mỗi đường biên được lưu trữ dưới dạng một vector của các điểm.  
**hierarchy** :  chứa thông tin về hình ảnh như số đường viền, xếp hạng các đường viền theo kích thước, trong ngoài, ..  
**mode** :  
CV\_RETR\_EXTERNAL : khi sử dựng cờ này nó chỉ lấy ra những đường biên bên ngoài, nhưng biên bên trong của vật thể bị loại bỏ.  
CV\_RETR\_LIST : Khi sử dụng cờ này nó lấy ra tất cả các đường viền tìm được.  
CV\_RETR\_CCOMP : khi sử dụng cờ này nó lấy tất cả những đường biên và chia nó làm 2 level, những đường biên bên ngoài đối tượng, và những đường biên bên trong đối tượng.  
CV\_RETR\_TREE : khi sử dụng cờ này nó lấy tất cả các đường biên và tạo ra một hệ thống phân cấp đầy đủ của những đường lồng nhau.  
**method** :  
CV\_CHAIN\_APPROX\_NONE : sử dụng cờ này sẽ lưu trữ tất cả các điểm của đường viền .  
CV\_CHAIN\_APPROX\_SIMPLE : Ví dụ : một hình chữ nhật sẽ được mã hoá bằng toạ độ của 4 đỉnh.  
CV\_CHAIN\_APPROX\_TC89\_L1 or CV\_CHAIN\_APPROX\_TC89\_KCOS :  Áp dụng thuật toán xấp xỉ Tech-Chin.



Trong ảnh, những đường màu hồng là đường contour bao quanh vật thể, nhưng vì có quá nhiều đường bao quanh các vật thể không phải biển số nên chúng ta sẽ áp dụng những đặc trưng riêng về tỉ lệ cao/rộng, diện tích trong khung hình cố định như ở mục 2.1 để lọc ra đúng biển số cần tìm.

### 3.10.2.Lọc biển số

Đầu tiên ta làm xấp xỉ contour thành một hình đa giác và chỉ lấy những đa giác nào chỉ có 4 cạnh. Nghĩa là lúc xấp xỉ contour bộ nhớ chỉ ghi nhớ vị trí các đỉnh của đa giác đó thành một mảng. Số cạnh của đa giác sẽ bằng số đỉnh và bằng chiều dài của mảng đó.

|  |  |
| --- | --- |
| Contour chưa xấp xỉ đa giác | Contour đã xấp xỉ đa giác |

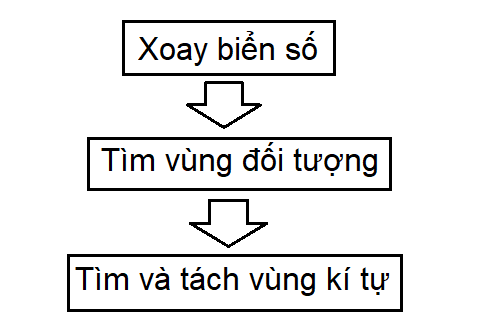
Tiếp theo ta tính toán tỉ lệ cao/rộng và diện tích của biển số phù hợp, sau đó ta lưu tất cả những biển số có trong hình dưới dạng tọa độ các đỉnh

Từ đây, ta cắt hình ảnh biển số từ các tọa độ vị trí đã biết để phục vụ cho mục đích tiếp theo “Tách các kí tự trong biển số”. Lưu ý ở đây ta cắt từ ảnh nhị phân luôn để máy tính xử lý nhanh hơn, tốn ít thời gian hơn.

## 3.11.Phân đoạn kí tự

### 3.11.1. Hướng giải quyết

Ở giai đoạn này có những bước chính sau: Xoay biển số để tăng khả năng nhận diện, Tìm tất cả các vùng kín cho là kí tự và lọc ra những kí tự đúng. Tách hình ảnh nhưng kí tự đó ra và đưa vào bộ nhận diện



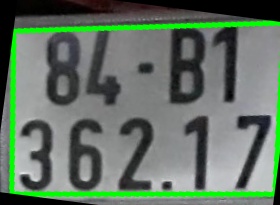
## 3.11.2.Xoay biển số

Khi chụp ảnh đầu vào, không phải lúc nào biển số cũng ở chính diện, có thể bị méo sang trái, sang phải, nghiêng góc dẫn đến nếu cứ sử dụng ảnh biển số đã cắt mà không điều chỉnh góc độ dẫn đến ảnh kí tự được cắt ra đưa vào bộ nhận diện rất dễ bị sai. Ví dụ giữa số 1 và số 7, số 2 và chữ Z, chữ B và số 8,...

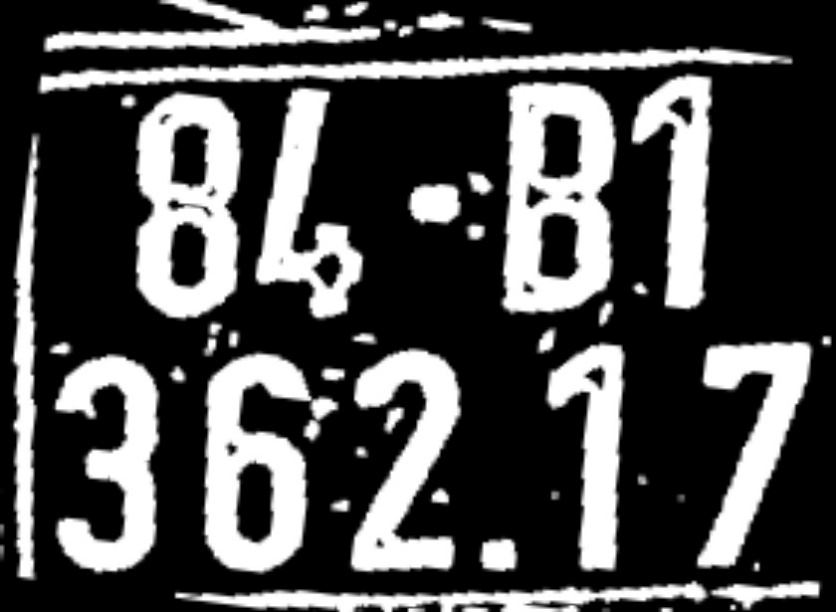


Phương pháp xoay ảnh em sử dụng ở đây là:

1. Lọc ra tọa độ 2 đỉnh A,B nằm dưới cùng của biển số
2. Từ 2 đỉnh có tọa độ lần lượt là A(x1, y1) và B(x2,y2) ta có thể tính được cạnh đối và cạnh kề của tam giác ABC
3. Ta tính được góc quay
4. Xoay ảnh theo góc quay đã tính. Nếu ngược lại điểm A nằm cao hơn điểm B ta cho góc quay âm



## 3.11.3.Tìm vùng đối tượng

Từ ảnh nhị phân, ta lại tìm contour cho các kí tự (phần màu trắng). Sau đó vẽ những hình chữ nhật bao quanh các kí tự đó. Tuy nhiên việc tìm contour này cũng bị nhiễu dẫn đến việc máy xử lý sai mà tìm ra những hình ảnh không phải kí tự. Ta sẽ áp dụng các đặc điểm về tỉ lệ chiều cao/rộng của kí tự, diện tích của kí tự so với biển số



Trong ảnh trên những đường màu vàng là đường contour và nếu so sánh với ảnh nhị phân 4.3 -1 thì có rất nhiều đường nhiễu như đường viền biển số, dấu gạch, dấu chấm... Sau khi đã áp dụng các điều kiện thì sẽ vẽ ra những hình chữ nhật màu xanh bao quanh các kí tự.

## 3.11.4.Tìm và tách kí tự

Sau khi đã nhận dạng từng kí tự bằng hình chữ nhật và cũng đã có tọa độ vị trí 4 đỉnh của hình đó, ta lúc này có thể cắt hình ảnh kí tự đó ra phục vụ cho giai đoạn sau “Nhận diện kí tự”. Lưu ý ở đây ta cắt ảnh nhị phân chứ không cắt từ ảnh gốc.



Hình 4.4 - 1 Ảnh kí tự sau khi cắt

# CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 4.1.Kết luận

Từ những kết quả thu được ở chương trên em nhận thấy phương pháp nhận diện biển số xe bằng xử lý ảnh và thuật toán KNN có những ưu và khuyết điểm sau:

Ưu điểm:

* Dễ cài đặt và sử dụng.
* Khá nhẹ nên máy tính với cấu hình yếu cũng có thể xử lý mượt mà so với các thuật toán khác như CNN, SVM.
* Phù hợp cho đối tượng sinh viên muốn tìm hiểu căn bản về xử lý ảnh hay trí tuệ nhân tạo.

Khuyết điểm:

* Khả năng nhận diện của KNN còn thấp, khi tập dữ liệu quá nhiều sẽ tăng thời gian xử lý vì phải quét hết tập dữ liệu train.
* Nhận diện kém với sự phản chiếu của biển số, sự di ảnh, chói sáng từ môi trường ngoài, những biển có phần chữ số không rõ ràng, với biển số xe ô tô

Vì vậy tốt nhất cần đặt camera cố định, với môi trường ánh sáng xung quanh được cài đặt trước. Phông nền cần hạn chế tối da nhưng chi tiết lóe sáng gây nhiễu. Phương pháp này vẫn còn cần sự giám sát của con người nhiều chứ chưa thể hoàn toàn tự động.

## 4.2.Hướng phát triển

Cần thay đổi thuật toán nhận diện KNN sang những thuật toán khác tinh vi và phức tạp hơn như CNN, SVM hoặc có thể sử dụng những bộ thư viện đã có sẵn trên thế giới như YOLO, YOLOv3 ....

Sử dụng camera chuyên dụng cho việc nhận diện biển số xe vì có khả năng chống chịu với sương mù, đêm tối, chói sáng,...

Sử dụng các thuật toán xử lý ảnh khác để xác định vị trí biển số tốt hơn như phương pháp biến đổi Hough để nhận diện đường thẳng, xác định bằng màu sắc, những thuật toán làm hạn chế sự di ảnh khi xe đang di chuyển.

Kết hợp với những chương trình khác để quản lý kho bãi, quản lý các phương tiện đang tham gia giao thông, tìm xe thất lạc, theo dấu,...

**TÀI LIỆU THAM KHẢO**

**[1] K-NEAREST NEIGHBORS:**

https://machinelearningcoban.com/2017/01/08/knn/

**[2] Support Vector Machines**

https://www.geeksforgeeks.org/introduction-to-support-vector-machines-svm/